

Termékstruktúrán alapuló fejlesztési folyamatmodellezés és –optimalás

Product Structure Based Process Modelling and Optimisation

Groma István, Dr. Bercsey Tibor

Budapesti Műszaki és Gazdaságtudományi Egyetem, Gépszerkezettani Intézet

Abstract

This paper deals with a modeling and optimization method that is capable of the product structure based optimization of design processes. The processes were modeled with Design Structure Matrix. Since the precedence task is a whole NP one, guided stochastic search techniques were applied. The GA was used to multi-object optimize the time and cost of design processes. The research supported by application OTKA T032474.

Összefoglaló

A cikk a tervezési folyamatok termékstruktúra alapú modellezését és optimalását mutatja be. A folyamatok modellezésére a Design Structure Mátrixot használtuk. Mivel a sorrendtervezési feladat bizonyítottan NP teljes feladatnak minősül, ezért egy sztochasztikus kereső eljárást alkalmaztunk az optimalási feladat megoldására. A GA-t a tervezési folyamatok idő és költség szerinti multi objekt optimalására. Ez a kutatómunka a T032474 OTKA keretei között zajlott.

Kulcsszavak

Design Structure Matrix (DSM), genetikai algoritmus (GA), tervezési folyamat, optimalás

1. Bevezetés

A vállalatok gazdasági sikerét a termék, és ezen keresztül a termék előállítási folyamat határozza meg. A termék előállítási folyamatban meghatározó szerepet játszik a konstrukciós tervezés, amely hatékonyságának megítélésében a minőségi követelményeken túlmenően a legfontosabb szempontok közé tartozik a tervezéshez szükséges idő és költség ráfordítás, valamint az ezekkel szorosan összefüggő optimális erőforrás kihasználás.

A konstrukciós tervezési folyamat lefutása, idő, költség és erőforrás igénye nagymértékben függ a termék és a konstrukció fajtájától, a termék bonyolultságától és szerkezeti felépítésétől, struktúrájától [4]. Ezért a konstrukciós folyamat tervezésénél elengedhetetlen a termék ezen sajátosságainak figyelembevétele [5], amelyek a VDI 2221 [13] tervezési folyamat modelljét alapul véve összességében, csak a koncepció képzést követően állnak rendelkezésünkre. Ebben a fázisban tisztázódik, hogy mely termék koncepció kerül a tervezés és kivitelezés fázisába, valamint a menedzsment és a piac által meghatározott előzetes költség és időterv. A tervezés és kivitelezés folyamatához szükséges erőforrásokat a fenti paraméterekhez kell illeszteni.

Az erőforrások azonban több projekthez is hozzá lehetnek rendelve és időben, költségben korlátozottak. Ezért napjainkban a középpontba került egy olyan termékfejlesztési folyamat és módszertan kidolgozása, amelyben a műszaki-technikai problémák megoldása mellett megfelelő hangsúlyt kap a fejlesztési-tervezési folyamat szervezése, az idő, erőforrás, kapacitás, költség valamint az információ folyam tervezése [7]. A bázist ehhez a komplex feladathoz a Design Structure Matrix (DSM) biztosítja.

2. Design Structure Matrix

Stewart [10] volt az első, aki információs és szervezeti kapcsolatok leírására a DSM-et alkalmazta. A módszer azonban csak számítógépesítése [9] után vált igazán ismerté és jól használhatóvá.

A DSM módszer azon alapszik, hogy a részegységek tervezési folyamatai közötti kapcsolat alapján a tevékenységek sorrendje átrendezhető, ezzel átláthatóvá és rövidebbé (olcsóbbá) téve a teljes folyamatot.

A tervezendő termék szerkezeti elemei közötti kapcsolatok leírásához egy mátrixot használunk, melyre a következők érvényesek:

A termék A_i ($i=1,2,\dots,n$) fő szerkezeti elemei megadják az 1. ábrán látható mátrixot. A diagonál elemei önmagukat reprezentálják, azaz $a_{ij}=0$ ($i=j$). A mátrix többi eleme a főbb szerkezeti elemek közötti kapcsolatok ábrázolására szolgál. Ha A_i struktúra elem információt ad A_j -nek, akkor $a_{ij}=1$, egyébként $a_{ij}=0$, ami azt jelenti, hogy az A_i és A_j elemek között nincs kapcsolat. Ha a mátrix egyik elemére igaz, hogy $a_{ij}=1$, és $i < j$, akkor ez az elem a diagonál felett van és előrecsatolt (feed forward) kapcsolatot jelent, amennyiben $i > j$, az elem a diagonál alatt van, visszacsatolást (feedback), vagy ciklust jelent. Ciklus esetén megadható az aktuális sorrend szerinti vélt ciklusok száma.

A mátrix elemeihez hozzá lehet rendelni további számos információt, amely a DSM módszer alkalmazhatóságát növeli. A következőkben bemutatásra kerülő eljárás jelenlegi formájában a menedzsment által megszabott költséget és időt kezeli.

A mátrix felvételekor a sorrend miatt előfordulhat, hogy a visszacsatolások száma és „mérete” (több elemet foglal magába) nagy, ez természetes módon azt jelenti, hogy nagy költséggel és időtöbblettel járnak. Ilyenek találhatóak például az 1. ábrán: 20-8, 18-5, sor-oszlop kombinációban.

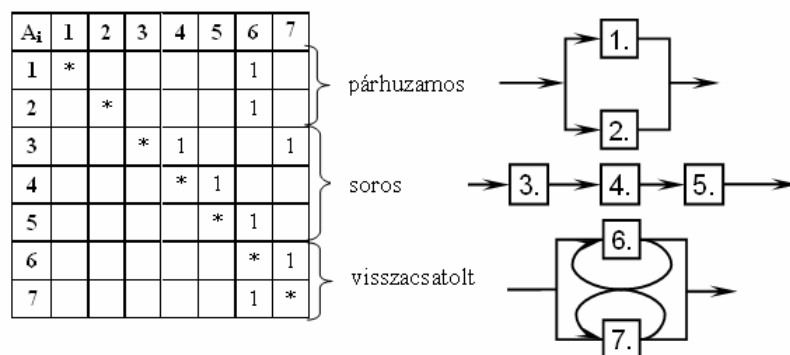
Az információfolyam szempontjából nagyon hátrányos, ha a ciklusok kereszteződnek. Ezek a tervezés során az információ redundancia, de leginkább az információk bizonytalansága miatt szintén költség többletet, és kaotikus eseményeket eredményezhetnek. Ilyenek például: 20-14 ciklus összefonódása a 17-10 ciklussal, (1. ábra).

A cél, hogy a DSM elemeinek egy olyan sorrendjét állítsuk elő, amelyben a lehetőség szerint minimális a visszacsatolások és kereszteződések száma, a költség és idő értékek csökkentése mellett. Ez a feladat egy olyan optimáló algoritmust igényel, amely robusztus feladatok megoldására is alkalmas, és képes egyszerre több súlyozott cél szerinti gyors optimálásra. Ennek megfelelően választottuk a genetikai algoritmusokat az optimálás végrehajtására.

Sz.e.	T	K	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	30	30									1													
2	40	20	8																					
3	10	50																		1				
4	10	50														1								
5	10	50								1														
6	50	10	4	2						1												1	1	
7	40	20														1								
8	50	10	8						6	2														
9	20	40			7				2															
10	20	40							8							1								
11	40	20						8																
12	30	30											5											
13	30	30												3										1
14	20	40												6										
15	30	30							4						8									
16	20	40													6		7							
17	30	30									8						7		8					
18	40	20				6												8		6				
19	50	10																	6		2			
20	40	20							7						4					2				
21	20	40															8							
22	20	40																						

1. ábra
DSM

A mátrixban szereplő kapcsolatok kifejtethők blokk diagrammá a 2. ábra alapján.



2. ábra
Kapcsolatok ábrázolása blokk diagrammban

3. Genetikai algoritmus

A '60-as évek elején John Holland és Ingo Rechenberg egymástól függetlenül megkísérelt egy olyan algoritmust kidolgozni, mely a természetes evolúció mechanizmusán alapszik. Rechenberg eredménye „evolúciós stratégia” néven vált ismertté, Hollandé pedig a „genetikus algoritmus” nevet viselte. A genetikus algoritmusok hatékony eszközök a diszkrét optimumkereséshez. A genetikus algoritmusok nem minden esetben közvetlenül az optimalizációs feladat paramétereivel dolgoznak, hanem e paramétereket reprezentáló bináris kódokkal operálnak, amelyek ugyan numerikus problémáknál jól adaptálhatók, de sorrendoptimalizálásnál használhatatlanok [6], [8].

Tehát egy terméket egy olyan jelvektorral ábrázolunk, ami megfelel a természetes kromoszómának. A jelvektort a biológiai megfeleltetés okán kromoszómának hívjuk. A keresési tér minden pontját, amely az összes termékvariánst tartalmazza, ábrázolni lehet egy kromoszómával. A genetikus algoritmus alapú keresés egy véletlenül előállított indulási populációból indul ki.

3.1. Kódolás

A genetikai algoritmusok legtöbbször nem direkt, hanem kódolt formában (pl.: bináris vagy gray kódolás) dolgoznak az optimalizandó paraméterekkel, de sorrendoptimalizálás esetén ez nem célravezető [2], [3]. Esetünkben a gén (egy adott sorrend, a keresési tér egy megoldása) a szerkezeti elemek sorszámából (kromoszóma) áll, kódolatlan formában.

3.2. Kiválasztás

A véletlenszerűen előállított kezdeti populáció egyedei közül, a kiértékelés után, két fajta szelekciós eljárás hatékonyságát vizsgáltuk. Az egyik a „Better Half” szelekció [1], ahol a populáció jobbik felét választjuk a további genetikai operátorok működtetésére. A másik típus a „Tournament,” szelekció [12], ahol véletlenszerűen kiválasztott két egyed rátermettségét hasonlítjuk össze, és a jobb rátermettségű egyed jut tovább.

3.3. Keresztezés

A keresztezés során kicserélődik a genetikai információ két egyed között, és létrejön egy új egyed. Az algoritmus a keresztezést ez esetben az úgynevezett pozíció bázisú (Position based crossover) keresztezéssel végzi el [11]. Ez azt jelenti, hogy az egyik szülőelemből véletlenszerűen kiválaszt az algoritmus egyes kromoszómákat, majd ezeket a kiválasztás helye szerint átírja az utód génjébe. A maradék helyeket a másik szülő elemeivel tölti fel úgy, hogy végigellenőrzi a sorrendet, és az első, amelyik még nem szerepel az utódban, átkerül az utód első üres kromoszóma helyére.

3.4. Mutáció

A mutáció során a keresztezés által előállított utódban véletlenszerűen kiválaszt az algoritmus két kromoszómát, és az ott lévő értékeket felcseréli. Ez a sorrend alapú mutáció [11].

3.5. Kiértékelés

Tervezési folyamatok optimalása megköveteli a költség és idő szerinti optimalást. Ennek megfelelően a kiértékelési eljárást úgy alakítottuk ki, hogy képes legyen az idők és költségek együttes figyelembevételére. A fejlesztés során ügyeltünk arra, hogy a vevői követelmények e két cél tekintetében és fontosságában különböző lehet. Így mindkét paraméterre bevezettünk súlyozási tényezőket. Ezzel beállítható, hogy a költség, vagy az idő cél a fontosabb.

Azért, hogy a modell jobban közelítse a valóságot, a ciklusokra bevezettünk egy további tanulási tényezőt (learning rate). Használatát az indokolja, hogy a ciklusokban szereplő, többször befutott tervezési lépések egyre kevesebb időt és költséget igényelnek, hiszen a megoldásukhoz szükséges apriori tudás is egyre kevesebb.

A célfüggvény általános definíciója (1):

$$f(\underline{wt}, \underline{wi}, \underline{tt}, \underline{ct}_{\langle s \rangle}, TI) := [\underline{wt}_1 \quad \dots \quad \underline{wt}_n] \times \begin{bmatrix} \frac{1-tt_1^{ct_1}}{1-tt_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1-tt_n^{ct_n}}{1-tt_n} \end{bmatrix} \times \begin{matrix} T \\ \begin{bmatrix} t_1.i_1 & \dots & t_1.i_m \\ \vdots & & \vdots \\ t_n.i_1 & \dots & t_n.i_m \end{bmatrix} \end{matrix} \times \begin{matrix} \begin{bmatrix} wi_1 \\ \vdots \\ wi_m \end{bmatrix} \end{matrix} \quad (1)$$

T	A szerkezeti elemek halmaza.
$n \in N_1 : T = n$	A szerkezeti elemek száma.
$m \in N_1$	Az értékelésnél releváns indikátorok száma.
$\langle s \rangle$	A szerkezeti elemek egy gyártási sorrendje.
$\underline{wt} \in R^n$	Az egyes szerkezeti elemek súlya a célfüggvényben.
$\underline{wi} \in R^m$	Az egyes indikátorok súlya a célfüggvényben.
$\underline{tt} \in [0,1]^n$	A szerkezeti elemekre vonatkoztatott tanulási ráták vektora.
$\underline{ct}_{\langle s \rangle} \in N_0^n$	A szerkezeti elemek adott gyártási sorrend mellett, ismételt előállítási számát tartalmazó vektor.
$TI \in R^{n \times m}$	A szerkezeti elemekhez tartozó indikátorértékek mátrixa.

A célfüggvény belső, számításakor használt alakja (2):

$$f(\underline{ct}_{\langle s \rangle}) = \underline{l}^T \times \left(I - \begin{bmatrix} tt_1^{ct_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & tt_n^{ct_n} \end{bmatrix} \right) \times \underline{r} \quad (2)$$

$\underline{l}, \underline{r} \in R^n$	Célfüggvény számításához tartozó segédvektorok.
$I \in R^{n \times n}$	Egységmátrix.

A feladat megfogalmazás (3):

$$\langle s_0 \rangle \in Perm(T) : f(\underline{ct}_{\langle s_0 \rangle}) \approx \min_{\langle s \rangle} f(\underline{ct}_{\langle s \rangle}) \quad (3)$$

$$|Perm(T)| = n!$$

Tehát, keressük azt a sorrendet, amelynél az átfutási idő és a folyamat teljes költsége minimális.

3.6. Algoritmus paraméterek

Az algoritmus futtatásához a következő paramétereket használtuk:

- mutáció valószínűsége: 0,4
- kereszteződés valószínűsége: 0,6
- tanulási ráta: 0,95

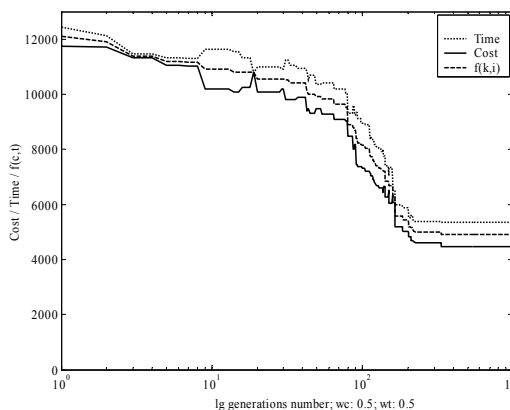
- populáció nagyság: 20
- generációk száma: 1000

4. Eredmények

A vizsgált feladat 22 szerkezeti elemből áll, 24 db visszacsatolást és 16 kereszteződést tartalmaz. A kiindulási sorrend 12066 időegységet és 12020 költségegységet igényel. A kiindulási mátrixot a 4. ábra mutatja, ahol S.E. a szerkezeti elem száma, T a tervezés ideje, C a tervezés költsége.

Az optimalás során a súlyozási tényezők hatását vizsgáltuk. Az első optimalás során a súlyozás értékei azonosak voltak, azaz $w_c = w_t = 0,5$. Az optimalt sorrendet és az optimalás lefutását a 3. ábra mutatja. Az optimalással a költségigény 4476 egységre, az időigény 5359 egységre csökkent.

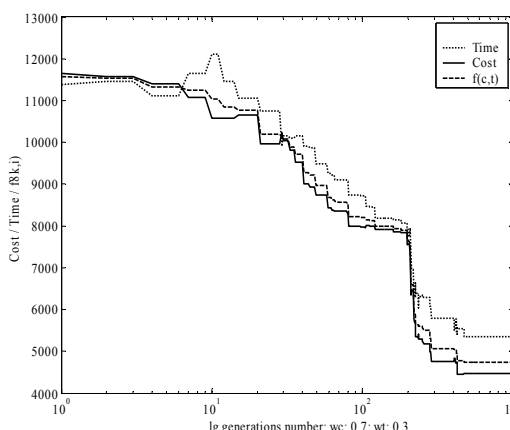
S.E.	T	C	12	11	6	3	21	16	17	18	19	20	10	15	5	13	14	8	7	9	2	1	4	22
12	30	30	1																					
11	40	20		1																				
6	50	10			1	1						1						1					1	
3	10	50								1														
21	20	40						1																
16	20	40					7						1											
17	30	30					7	8					1											
18	40	20					8	6						1										
19	50	10										2												
20	40	20											2					1	1					
10	20	40																1	1					
15	30	30																1	1					
5	10	50																	1					
13	30	30	3																	1				1
14	20	40														6								
8	50	10																	1	2	1			
7	40	20																1						
9	20	40																2						1
2	40	20																				1		
1	30	30																		1				
4	10	50																1						
22	20	40																						



3. ábra
Optimalt DSM 1, optimalás lefutása

A második optimalás során a súlyozás értékei a következők voltak: $w_c = 0,7$; $w_t = 0,3$. Az optimalt sorrendet és az optimalás lefutását a 4. ábra mutatja. Az optimalással a költségigény 4450 egységre, az időigény 5370 egységre csökkent.

S.E.	T	C	12	11	6	3	21	16	17	18	19	20	10	15	5	13	14	8	7	9	2	1	4	22
12	30	30	1																					
11	40	20		1																				
6	50	10			1	1						1							1					1
21	20	40						1																
3	10	50								1														
16	20	40						7						1										
17	30	30						7	8					1										
18	40	20						8	6						1									
19	50	10											2											
20	40	20																	1	1				
10	20	40																	1	1				
15	30	30																	1	1				
5	10	50																		1				
13	30	30	3																		1			1
14	20	40														6								
8	50	10																	1	2	1			
7	40	20																1						
9	20	40																	2					1
2	40	20																				1		
1	30	30																			1			
4	10	50																1						
22	20	40																						



4. ábra
Optimalt DSM 1, optimalás lefutása

Az algoritmus működéséhez meg kell jegyezni, hogy a célfüggvény alkalmas megválasztásával az összetartozó elemeket alfolyamatokba rendezte az algoritmus (rövidítve az átfutás idejét és csökkentve a költségeket), külön clustering algoritmus alkalmazása nélkül (3. ábra 12-13. és 14-22 elemek közötti rész). Összességében megállapítható, hogy a kidolgozott módszer lehetőséget ad a tervezési folyamatok átlátható optimalására. Továbbá teljesíti a szabványos rendszerekkel való együttműködést, az oda- és visszakonvertálást.

A rendszer hatékonyságát felvétele során bevezetett valószínűségek használatával szeretnénk növelni, így egy teljesen új termék tervezése során is képes a rendszer helyes megoldást adni, úgy, hogy a nagyon gyenge kapcsolatokat nem veszi figyelembe. Jelenleg a célfüggvény számítása során teljes emberórát számítunk. Ez ugyan a sorrend szempontjából optimális, de a folyamatot nem lehet időben elhelyezni. Ezért szükséges az optimált DSM kikonvertálása egy hálótervvé, amely már lehetővé teszi a valós átfutási idő kiszámítását. Itt egy további lehetőség nyílik arra, hogy az erőforrásokat a dátumozott folyamat, azaz projekt követelményeinek megfelelően lehessen az egyes feladatokhoz hozzárendelni.

Irodalmi hivatkozások

- [1] ABRAMSON, D., LEWIS, A., PEACHEY, T., FLETCHER, C.: An Automatic Design Optimization Tool and its Application to Computational Fluid Dynamics. Proc., ACM/IEEE SC2001 Conf., Denver, CO, 2001.
- [2] ALTUS, S., S.; KROO, I., M.; GAGE, P., J.: A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems, ASME Paper 95-141., 1995.
- [3] BLOEBAUM, C.L.: An Intelligent Decomposition Approach for Coupled Engineering Systems, proceedings of the Fourth AIAA/AF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, Cleveland, OH.. 1992.
- [4] DONALD, G., R.: Die neuen Werkzeuge der Produktentwicklung, Carl Hanser Verlag, München, Wien, 1998
- [5] EHRENSPIEL, K.: Integrierte Produktentwicklung: Methoden für Prozessorganisation, Produktentwicklung und Konstruktion, Carls Hansen Verlag München Wien, 1995.
- [6] HOLLAND, J.: Adaptation in natural and artificial systems, MIT Press, Cambridge, Mass 1975
- [7] MARCA, D. A.; MCGOWAN, C. L.: Structured analysis and design technique: SADT, McGraw-Hill, New York 1988
- [8] RECHENBERG, I.: Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipen der biologischen Evolution, Friedrich Frommann Verlag, Stuttgart 1973
- [9] ROGERS, JAMES L.: DeMAID/GA - An Enhanced Design Manager's Aid for Intelligent Decomposition, 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Seattle, WA , September 4-6, 1996a. AIAA paper No. 96-4157.
- [10] STEWARD, D., V.: SystemAnalysis and Management: Structure, Strategy and Design, Petrocelli Books Inc. 1981.
- [11] SYSWERDA, G.: Schedule Optimization Using Genetic Algorithms, Handbook of Genetic Algorithms, Van Nostran Reinhold, New York, 1990.
- [12] T. BACK. Generalized Convergence Models for Tournament- and Selection, In Larry J. Eshelman, editor, Proc. Sixth Int. Conf. on Genetic Algorithms, 2-8. Morgan Kaufmann Publishers, San Francisco, 1995.
- [13] VDI-RICHTLINIE 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte, Düsseldorf, VDI-Verlag 1986